

The Butterfly Effect

or

How Support for Complex Hardware may affect the uAPI

Michael Riesch

Media Summit 2026, Nice



COLLABORA

Open First

\$ whoami

- First steps in the kernel 2009-2013
- Full time kernel developer since 2020
- Joined Collabora in 2025
- mriesch @OFTC, @Libera
- michael.riesch@collabora.com
- Mostly linux-media, linux-rockchip, ...
- RK35xy video capture and camera support since 2021



COLLABORA

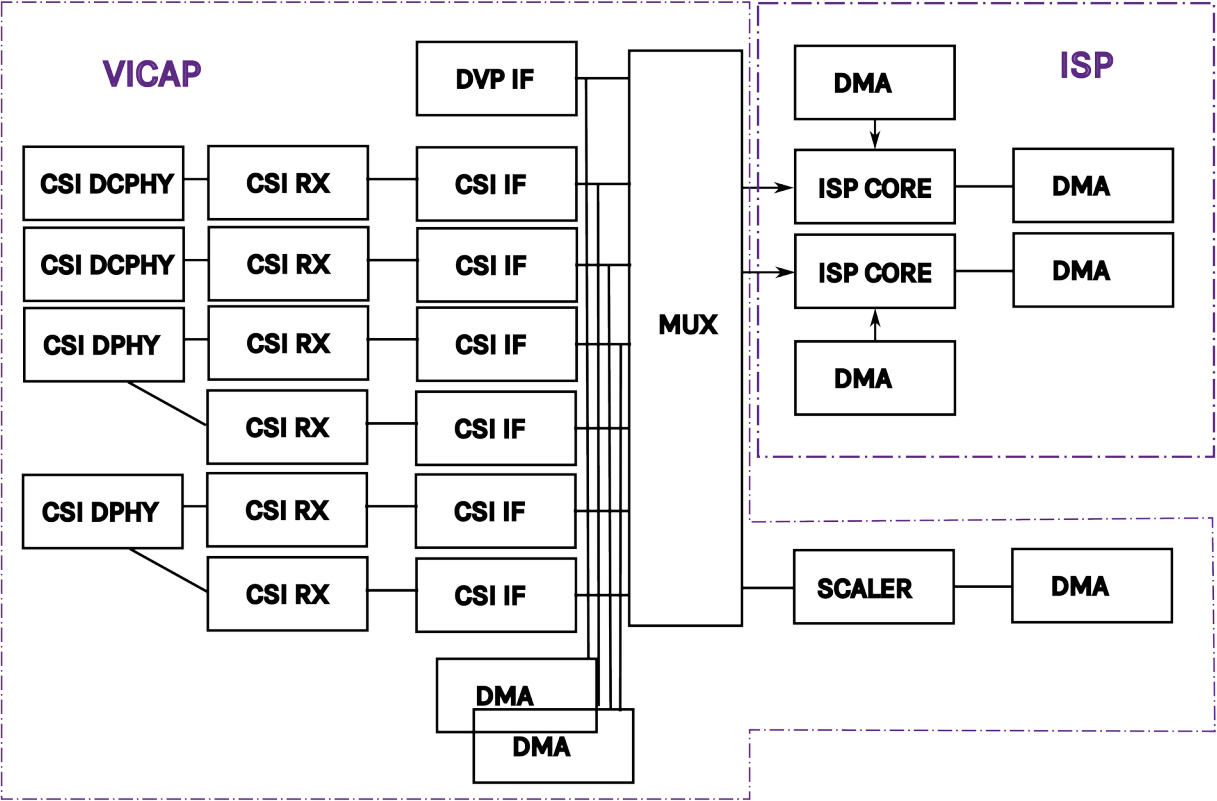
Open First



COLLABORA

RK35xy Video Capture and Camera Support

RK3588 VICAP and ISP



Status Matrix 1/2

⁰ v6.1

² v6.19

* should be easy™

¹ v6.18

³ v6.20 (w/o DT)

SoC	RK3566/68	RK3588	RK3576	RK3562
MIPI CSI-2 DPHY	DONE ⁰	DONE ¹	TODO	TODO
DPHY split mode	TODO	TODO	TODO	TODO
MIPI CSI-2 DCPHY	n/a	TODO	TODO	n/a
MIPI CSI-2 receiver	DONE ³	DONE ³	TODO*	TODO*
VICAP MIPI capture	DONE ²	WIP	TODO*	TODO*
VICAP DVP capture	DONE ²	TODO*	TODO*	n/a
VICAP MUX/TOISP	n/a	WIP	TODO	TODO
VICAP SCALER	n/a	TODO	TODO	TODO



COLLABORA

Open First



First stream with a Radxa ROCK 5B+ (RK3588) + Radxa Camera 4K (IMX415) via VICAP and ISP



Status Matrix 2/2

nothing easy™ here

SoC	RK3566/68	RK3588	RK3576	RK3562
ISP MIPI CSI-2 RX	WIP (paused)	n/a	n/a	n/a
ISP main/self path	WIP (paused)	WIP	TODO	TODO
ISP stats/params/3A	WIP (paused)	TODO WIP	TODO	TODO
ISP raw capture	TODO	n/a	n/a	n/a
ISP mem2mem	TODO	WIP	TODO	TODO
ISP HDR	TODO	TODO	TODO	TODO
ISP dual support	n/a	TODO	n/a	n/a
libcamera support	TODO	TODO WIP	TODO	TODO



COLLABORA

Open First



What is Required for the Next Steps?

Things to Discuss

- How to expose the HW using V4L2/MC?
 - Number of media graphs/media devices (/dev/mediaX)?
 - How to connect VICAP to the ISP in V4L2/MC?
 - How to enable mem2mem or inline ISP operation mode?
 - How to enable dual ISP operation mode?
- V4L2/MC features
 - Streams API for MIPI CSI-2 VC/DT: still experimental, needs testing
 - Multicontext MC (WIP) for time-division multiplex
 - Scheduler framework in V4L2/MC?
- ?

How to Enable Inline ISP Mode?

- VICAP can pass image data to the ISPs => inline mode
- Each ISP can process image data from memory and put the result back into memory => mem2mem mode
- How to switch between these two modes?
 - At runtime, no device tree magic (as \$vendor likes to do)
 - Additional MUX subdevice with two inputs (mem, vicap) and one output



How many Media Devices for RK3588?

- Both VICAP and the ISPs can work on their own, respectively.
- One media device for all of them? Or one for each VICAP + ISPO + ISP1?
- If one media device for all:
 - Which entity registers the media device?
 - Do we need a Rockchip Camera Subsystem phony driver?
 - Do we need one system-wide media device to rule them all?



How to Register the Subdevices?

- Usually there is a single entity that registers all the subdevices in the pipeline.
- Which entity would that be for the VICAP + ISPO + ISP1 combo?
- Possible solutions:
 - The Rockchip Camera Subsystem phony driver registers them all.
 - We modify the `v4l2_{async, device}_register_subdev` methods



This escalated quickly ;-)

- The latter caused a bit of discussion^{0,1}
 - `<pinchartl>` what you described sounded like a quick hack
 - `<pinchartl>` you instead need to **design** a solution
 - Dynamic Media Graph

⁰ https://linuxtv.org/irc/oftc/irclogger_log/linux-media?date=2026-02-17,Tue&sel=72#l68

¹ https://linuxtv.org/irc/oftc/irclogger_log/linux-media?date=2026-02-18,Wed&sel=14#l10



Dynamic Media Graph

- Media entities can appear/disappear/reappear at any time.
- There needs to be at least one uAPI change to provide notifications of change to user space.
- User space must be able to handle the changes.
- User space must be able to handle an incomplete graph.
- - *- kbingham screams in pain "fault tolerant v4l2"
- Indeed there is significant overlap with fault tolerant V4L2²

² <https://www.linuxtv.org/news.php?entry=2022-11-14-0.hverkuil>





What Else Would be Nice?

What Else?

<pinchartl> mriesch: you need to take a step back and look at the big picture

- Streams (WIP)
- Multicontext MC (WIP)
- Dependencies between processing steps, jobs (WIP)
- Scheduling (e.g., Time Division Multiplexing in ISPs)
- Internal pads (WIP)
- Metadata (WIP)
- Number of V4L2 (sub)devices???
- Atomic uAPI???
- Better support for 2D accelerators (e.g., RK3576 inline VPSS, RK3588 RGA3, ...)???
- ... ?
- Any thoughts?



So.. What Should We Do?

Extend the existing APIs (kAPI, uAPI)? → painful!

- `<sailus>` There are a few things that should be done here to get rid of the current restrictions we have: [...]
- `<hverkuil>` `mriesch`: sure, as long as it doesn't break existing drivers. And that's where IMHO the problem is. It's not in the kernel APIs, it's how to handle it in userspace. Right now **everything** relies on the fact that all hardware is present and working. And you can't break that. So how do you do this? That's what the RFC should explain.



So... What Should We Do?

Come up with something new → painful, too!

- Painful, but hopefully rewarding in the long run → designed for the purpose
- Are we able to move faster with a clean cut?
- Either way, it makes sense to think about what the final result could/should look like.

<pinchartl> now that we have mriesch volunteering things can finally move forward:-)

<mriesch> pinchartl: in principle yes, but the direction should be clear for that





Thank you!



COLLABORA

Open First